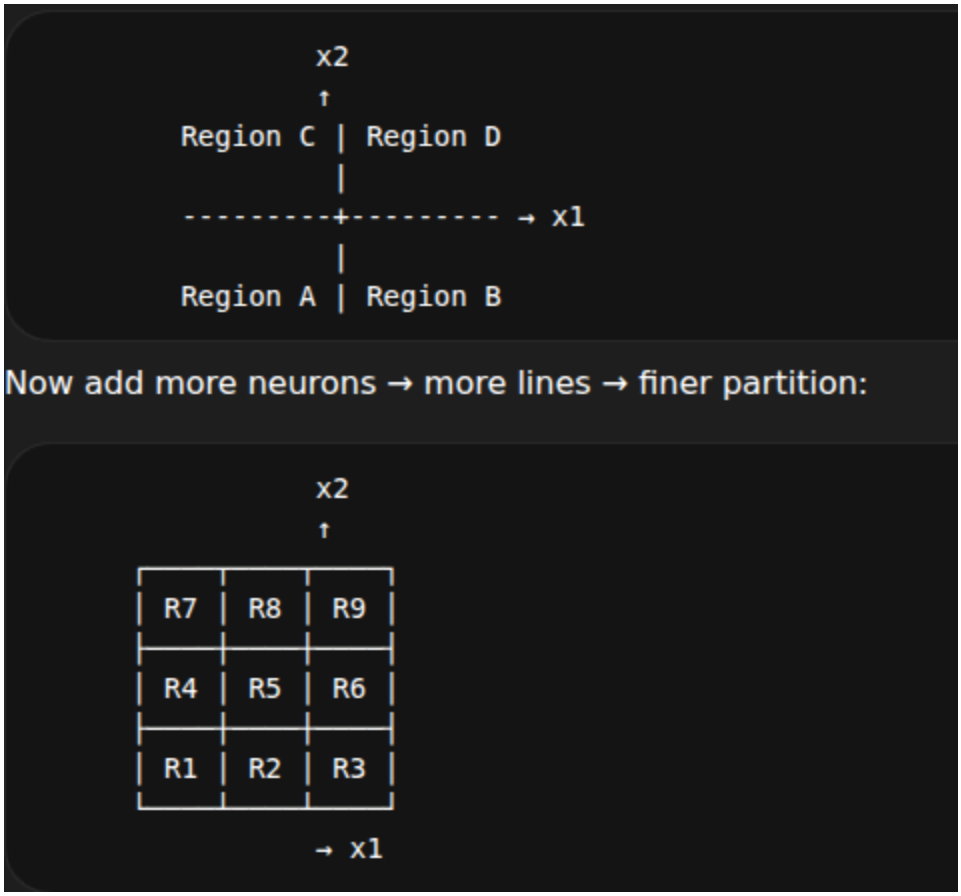
Here's a **region partition diagram** that shows how a ReLU network splits input space into regions, each with its own effective matrix $W_{eff(x)}$.

🔷 1. Input Space Partitioning (2D Example)

Imagine a 2D input $x = (x_1, x_2)$. Each ReLU neuron defines a boundary:

- A line where its pre-activation = 0
- This line splits space into:
    - ON region active = 1
    - OFF region inactive = 0

Partitioned Plane



Each region corresponds to a **fixed pattern of ReLU on/off states**.

🔷 2. What Each Region Represents

Inside any one region:

- Every ReLU decision is fixed (no switching)
- So each D(x) becomes a **constant diagonal matrix**

That means:

$W_{eff(x)} = W_3D_2W_2D_1W_1$

is **constant within that region**

🔷 3. Region → Matrix Mapping

Region R1 → Weff(1)
Region R2 → Weff(2)
Region R3 → Weff(3)
...
Region R9 → Weff(9)

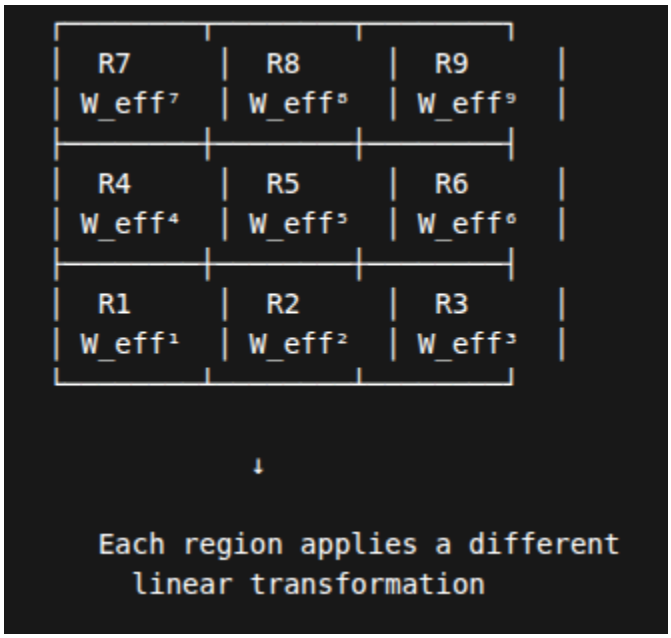
So the network behaves like:

if x ∈ R1 → y = W_eff(1)x
if x ∈ R2 → y = W_eff(2)x
if x ∈ R3 → y = W_eff(3)x
...

🔷 4. Full Diagram (Putting It Together)

INPUT SPACE (x)

Partitioned into regions by ReLU boundaries



🔷 5. Key Insight

The network is **piecewise linear**, not globally linear.

- Globally: nonlinear system
- Locally (inside a region): linear system

🔷 6. Why Capacity Explodes

If you have:

- n ReLU neurons → up to 2^n regions (upper bound)
- Multiple layers → **combinatorial explosion of regions**

So:

- Linear network → **1 region**
- ReLU network → **many regions**
- Each region → its own associative memory

🔷 7. Intuition Upgrade

You can now think of a ReLU network as:

Input space → partition → region selection → linear map → output

or even:

A giant IF-ELSE system:

if (pattern 1) → use matrix A
if (pattern 2) → use matrix B
...

🔷 8. Final One-Line Summary

A ReLU network creates a **partition of input space into regions**, where each region is governed by a different linear transformation, turning a single low-capacity linear map into a large collection of context-dependent associative memories.